**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**УНИВЕРСИТЕТ ИТМО**

**инистерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Прикладной информатики**

Направление подготовки **45.03.04 Интеллектуальные системы в гуманитарной сфере**

Образовательная программа[**Языковые модели и искусственный интеллект**](https://isu.ifmo.ru/pls/apex/f?p=2143:13:116963507483892::NO::EP_UCHEB_YEAR,EP_MEGAFACULTY,EP_FACULTY,EP_CATHEDRA,EP_GROUP,EP_FO,EP_KVAL,EP_UCHEB_PLAN:2024/2025,-1,725,-1,K3160,-1,-1,117277)

**К У Р С О В О Й   П Р О Е К Т**

Тема: «Сравнение библиотек сканирования штрих-кодов на Android»

Обучающийся: Маслов Иван Андреевич, К3162

Санкт-Петербург 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_heading=h.3dy6vkm)

[Актуальность 3](#_heading=h.1t3h5sf)

[Цели и задачи 3](#_heading=h.2s8eyo1)

[ГЛАВА 1. Основная идея проекта 5](#_heading=h.17dp8vu)

[ГЛАВА 2. Управление проектом 6](#_heading=h.3rdcrjn)

[2.1 Распределение ролей и обоснование 6](#_heading=h.2jxsxqh)

[2.2 Процесс ведения проекта 6](#_heading=h.z337ya)

[ГЛАВА 3. Выполнение проекта. 8](#_heading=h.3j2qqm3)

[3.1 Индивидуальные задачи 8](#_heading=h.1y810tw)

[3.2 Изучение принципов генерации штрих-кодов 8](#_heading=h.4i7ojhp)

[3.3 Изучение Python-библиотек для генерации штрих-кодов. 9](#_heading=h.2xcytpi)

[3.4 Разработка инструмента для генерации штрих-кодов 10](#_heading=h.1ci93xb)

[3.5 Оценка проделанной работы 13](#_heading=h.3whwml4)

[ЗАКЛЮЧЕНИЕ 14](#_heading=h.2bn6wsx)

[Выводы по работе 14](#_heading=h.qsh70q)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_heading=h.3as4poj)

[ПРИЛОЖЕНИЕ 1. ТЗ 16](#_heading=h.1pxezwc)

## 

## **ВВЕДЕНИЕ**

### **Актуальность**

Штрих-коды являются неотъемлемой частью работы большинства современных предприятий в области розничной и оптовой торговли. Они используются на всех этапах производства: для упрощения идентификации товара на этапе распределения, для автоматизации бизнес-процессов и логистики на этапе хранения и инвентаризации.

### **Цели и задачи**

Цель проекта – Исследовать существующие библиотеки сканирования штрих-кодов для мобильных устройств, провести сравнение и анализ, используя синтетические данные

Задачи:

1. Изучить теорию штрихкодирования и применение в различных бизнес-процессах,
2. Определить, какие форматы кодов часто используются в различных отраслях экономики,
3. Изучить IDE Android Studio / IntelliJ IDEA. Установить необходимое для работы вспомогательное ПО,
4. Изучить работу с файлами и изображениями в Java/Android и права доступа,
5. Разработать инструмент для генерации штрих-кодов различных форматов без дефектов (форматы ШК будут определены аналитиками в ходе работы),
6. Реализовать фильтры для наложения различных дефектов на изображение со штрих-кодом. Необходимые фильтры необходимо продумать совместно с аналитиками,
7. Разработать приложение для сканирования штрих-кодов с использованием библиотеки Zxing,
8. Разработать приложение для сканирования штрих-кодов с использованием библиотеки Google ML Kit,
9. Разработать приложение для сканирования штрих-кодов с использованием библиотеки Dynamsoft Barcode Reader,
10. Разработать приложение для сканирования штрих-кодов с использованием библиотеки Scandit,
11. Провести сравнительный анализ open-source библиотек, оформить таблицу с результатами,
12. Провести сравнительный анализ коммерческих библиотек, оформить таблицу с результатами.

## **ГЛАВА 1. Основная идея проекта**

Существует множество разных стандартов штрих-кодов, среди которых наиболее актуальными в реалиях российского рынке на данный момент являются EAN-13 – для маркировки товаров в розничной торговле, QR – для хранения большего объема данных, например, при складировании большого количества товаров и DataMatrix – для обязательной маркировки некоторых типов товаров по национальной системе «Честный знак». В связи с этим возникает потребность в многофункциональном сканере, предназначенном для сканирования штрих-кодов нескольких стандартов.

Штрих-коды по мере хранения могут повреждаться, возможны дефекты печати, освящение в помещении может быть неоднородным. В связи с этим возникает также потребность в сканерах штрих-кодов, способных корректно считать данные с поврежденного штрих-кода.

Специальные устройства для сканирования штрих-кодов являются дополнительной статьей расхода, относительно небольшие предприятия часто не могут позволить себе достаточное количество устройств, что приводит к снижению замедлению бизнес-процессов. Решением этой проблемы является использование специализированных приложений-сканеров на мобильных устройствах.

Таким образом, суть проекта заключается в разработке метрик и сравнении с их помощью существующих android-библиотек для сканирования штрих-кодов.

## **ГЛАВА 2. Управление проектом**

### **Распределение ролей и обоснование**

Следующим образом, были распределены роли внутри команды:

* Руководитель – Комаров Георгий Юрьевич
* Аналитики – Слиозберг Владимир Владимирович, Нгуен Хыу Жанг,
* Python разработчики – Маслов Иван Андреевич, Абакунов Кирилл Вячеславович,
* Мобильные разработчики – Розметов Джалолиддин, Черепня Ярослав Игоревич.

Для определения роли учитывались индивидуальные особенности: навыки, предыдущий опыт и личные предпочтения. Каждая роль отведена паре человек, так как в проекте некоторые задачи объемнее других, и их решение может потребовать больше времени. Кроме того, объединение усилий помогает участникам развить командные навыки и обращаться за помощью к коллегам в случае возникновения трудностей.

### **2.2 Процесс ведения проекта**

Для связи с командой использовался чат в Telegram, также для кооперации разработчиков был создан Gitlab-репозиторий.

Для коммуникации с командой также проводились регулярные групповые звонки на базе платформы google meet, во время которых участники отчитывались о состоянии поставленных перед ними задач, выстраивались дальнейшие планы и стратегии работы.

Руководитель брал на себя ответственность за взаимодействие членов команд из разных подгрупп, вносил корректировки в планы деятельности членов команды, выдавал задачи и разделял их на подзадачи. Со своей задачей руководитель справился отлично.

Ведение проекта производилось в сервисе ITMO Learning Projects. Задачи были распределены по ролям в команде, в рамках каждого спринта были проставлены сроки и назначены ответственные исполнители. (см. рисунок 1)

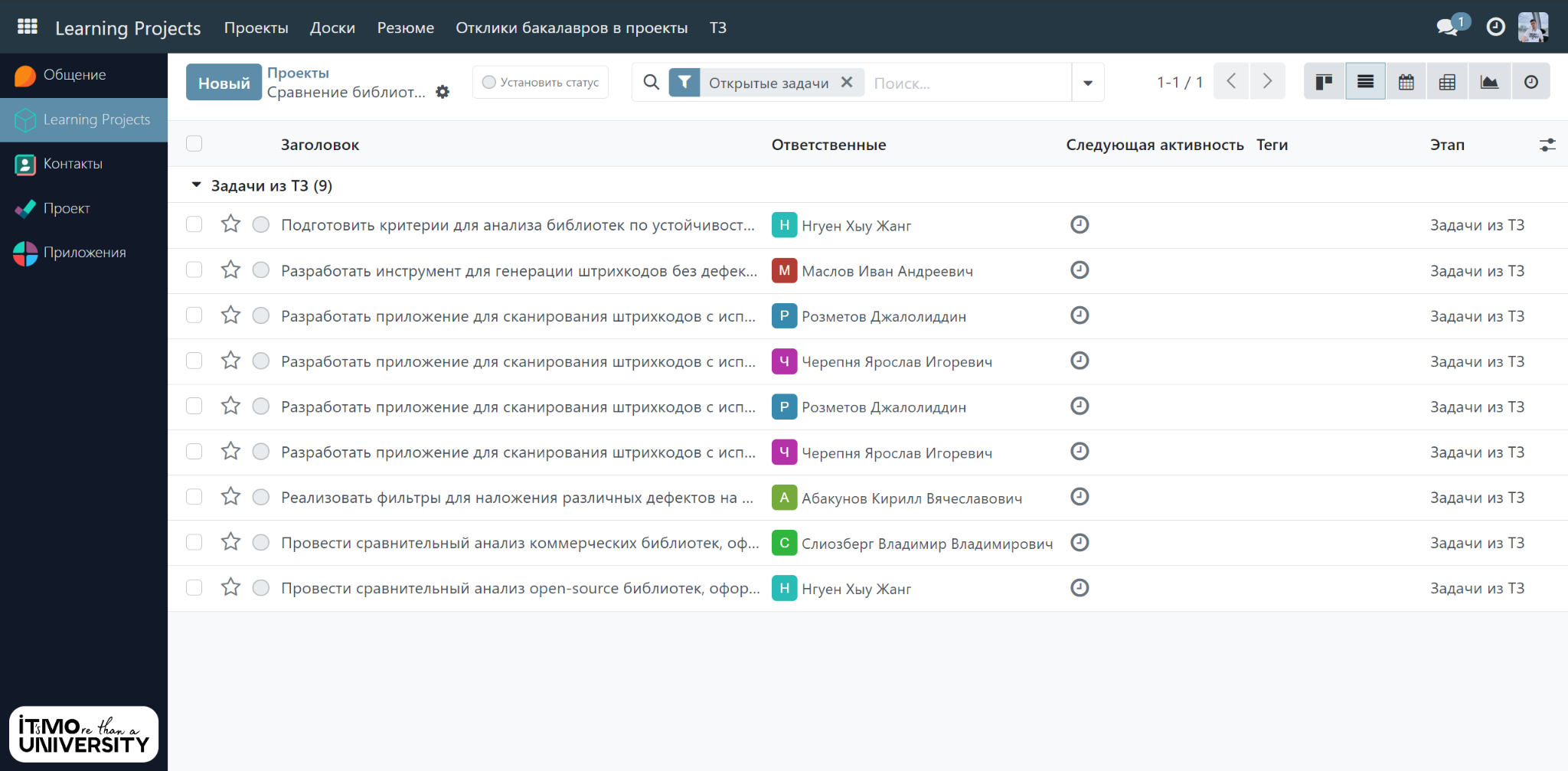


Рисунок 1 – ведение проекта в ITMO Learning Projects

## **ГЛАВА 3. Выполнение проекта**.

### **3.1 Индивидуальные задачи**

В рамках проекта я выполнял роль python-разработчика. Передо мной были поставлены следующие задачи:

1. Изучить принципы генерации штрих кодов форматов EAN13, QR, DataMatrix,
2. Изучить существующие Python-библиотеки для генерации штрих-кодов, выбрать среди них оптимальную,
3. Разработать инструмент для генерации неповрежденных штрих-кодов нужных стандартов.

### **3.2 Изучение принципов генерации штрих-кодов**

Основная идея кодирования информации с помощью штрих-кодов – представление нулей и единиц двоичного кода цветными элементами с разным альбедо – отражающим коэффициентом. Устройства-сканеры направляют на поверхность штрих-кода равномерный поток света и считывают разную интенсивность отраженного света, и в зависимости от нее определяют значение нужного соответствующего бита.

Штрих-код состоит из информационной и вспомогательных частей. На информационной непосредственно кодируется информация, вспомогательная часть, в свою очередь, содержит маркеры масштаба и ориентации штрих-кода, также может обеспечивать помехоустойчивость кода (например, штрих-коды формата EAN-13 содержат в конце контрольную сумму закодированных цифр).

Штрих-коды бывают черно-белыми и цветными. Последние менее популярны, так как, хотя и способны кодировать больший объем информации за единицу площади, требуют более точного оборудования.

Штрих-коды могут быть одномерными или двумерными. Первые кодируют информацию с помощью полос разных цветов, вторые – с помощью разноцветных пикселей. И первые, и вторые довольно популярны, так как двумерные кодируют значительно больше информации за единицу площади, а вторые требуют значительно более простого оборудования.

### **3.3 Изучение Python-библиотек для генерации штрих-кодов.**

Для наиболее оптимального использования ресурсов, было решено писать генератор штрих-кодов с помощью существующего специализированного модуля. Были найдены и рассмотрены следующие библиотеки Python:

1. Barcode,
2. Qrcode,
3. Asposebarcode,
4. Aspose-barcode,
5. Treepoem.

Библиотека Barcode предназначена для генерации одномерных штрих-кодов множества различных стандартов и отлично справляется со своей задачей: генерирует изображения формата jpeg и png хорошего качества. Не использует классы. Значительным плюсом также является наличие официальной документации. Однако, так как из выявленных в процессе анализа штрих-кодов только EAN-13 является одномерным, для выполнения задачи ее недостаточно.

Библиотека Qrcode предназначена для генерации штрих-кодов одноименного формата, поддерживает генерацию jpeg и png. Не имееи официальной документацию. В связи с популярностью библиотеки, однако, существует множество образовательных материалов по ее применению. Аналогично предыдущей библиотеке, для реализации идеи проекта ее недостаточно.

Библиотека Asposebarcode предназначена для генерации как одномерных, так и двумерных штрих-кодов разных стандартов. Не имеет официальной документации. При использовании Pycharm импортирование библиотеки вызывает ошибку, причина которой так и не была выявлена. В связи с этим, было решено отказаться от использования данной библиотеки.

Библиотека Aspose-barcode предоставляет инструменты для генерации как одномерных, так и двумерных штрих-кодов разных стандартов**.** У модуля есть официальная документация. Обширно использует классы.Библиотека платная, однако существует бесплатная пробная версия, сохраняющая весь функционал, но накладывающая на сгенерированные изображения вотермарку в виде серой диагональной линии. От использования пробной версии было решено отказаться, так как помехи могли значительно повлиять на точность дальнейшего исследования.

Библиотека Treepoem, как и Aspose-barcode предоставляет инструменты для генерации как одномерных, так и двумерных штрих-кодов разных стандартов**.** Документация отсутствует. Модуль имеет встроенную зависимость от другой библиотеке Ghostscript.

### **3.4 Разработка инструмента для генерации штрих-кодов**

Реализация генерации штрих-кодов с помощью данной библиотеки была достаточно прямолинейна, так как у нее есть официальная документация. Проблем при установке и импортировании модуля не возникло.

Первая версия инструмента для генерации неповрежденных штрих-кодов была написана с помощью библиотеки Aspose-barcode. На рисунке 2 представлен код генератора.

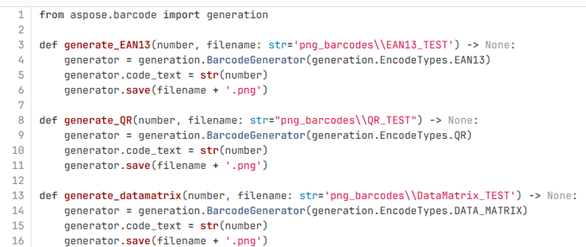


Рисунок 2 – генерация с Aspose-barcode.

Так как мы использовали бесплатную пробную версию платной библиотеки, на сгенерированных с ее помощью штрих-кодах оставалась вотермарка в виде серой линии, которая оставляла штрих-коды читаемыми, но могла значительно повлиять на точность дальнейшего исследования.

Компания, выпускающая Aspose-barcode предлагает возможность для временного получения полной версии модуля, но только для коммерческих организаций в целях тестирования и только на 30 дней. Хотя этого срока для завершения проекта вполне достаточно, в связи с трудностью получения полной версии и отсутствием гарантий ее получения, было решено отказаться от использования этой библиотеки.

В связи с этим было принято решение переписать инструмент для генерации на основании модуля Treepoem. Модуль обладает встроенной зависимостью от библиотеки Ghostscript.

При установке библиотеки напрямую через терминал с помощью pip install, модуль Ghostscript отображается в виртуальном окружении проекта, но вызывает ошибку при импортировании. После установки модуля в виртуальное окружение проекта, модуль корректно импортируется, однако Treepoem по-прежнему вызывает ошибку, так как не может найти к нему путь. Проблема решается размещением файлов модуля непосредственно в корневой папке проекта.

С помощью виртуальной машины с установленной на нее Ubuntu впоследствии было выявлено, что все сложности с библиотекой возникают из-за использования операционной системы Windows, и на устройствах с Unix-подобными ОС проблем не возникает.

На рисунке 3 представлен код второй версии генератора.

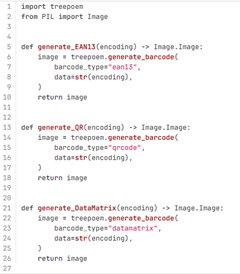


Рисунок 3–Генерация штрих-кодов с Treepoem

Treepoem также не имеет полной официальной документации. В связи с этим, для того чтобы разобраться с синтаксисом и функционалом модуля обширно использовалась команда dir(). Также, в официальных источниках не представлен полный перечень поддерживаемых кодировок. Тип кодировки в рамках этого модуля реализован так: параметр BarcodeType принимает на вход строковое значение – название кодировки как ключ словаря barcode\_types и выводит соответствующее ключу значение словаря – собственно кодировку. Выведя непосредственно этот словарь, можно увидеть, что модуль поддерживает все интересующие нас кодировки и формат, в котором их необходимо указывать при генерации штрих-кодов.

По завершении разработки, код был отправлен в Gitlab-репозиторий.

### **3.5 Оценка проделанной работы**

Я выполнил все поставленные перед собой задачи. В ходе проекта я близко взаимодействовал в основном со вторым Python-разработчиком и руководителем, коммуникация была крайне продуктивной.

Руководитель был всегда готов ответить на возникающие вопросы, на мой взгляд, он полностью справился со своей задачей.

Моя работа была скорее спонтанной, нежели планомерной, так как работу над проектом мне (как и всем участникам проекта), необходимо было совмещать с другой учебой и внеучебной деятельностью.

Я стал больше ценить уже имеющиеся у меня навыки. Например, знание английского языка очень помогло мне в поиске информации по теме и при изучении документации.

Выполнение проекта помогло мне развить навыки работы в команде, поиска информации, тайм-менеджмента, непосредственно аналитики и программирования.

## 

## **ЗАКЛЮЧЕНИЕ**

### **Выводы по работе**

В результате выполненной работы были созданы метрики и инструмент для сравнения библиотек, на основании синтетических данных. С помощью сгенерированного датасета была проведена аналитическая работа: был проведен сравнительный анализ четырех библиотек для сканирования штрих-кодов, двух open-source и двух с закрытым исходным кодом.

Все поставленные перед нами задачи были выполнены, цель достигнута.

Мой вклад в работу заключается в создании инструмента для создания неискаженных штрих-кодов и генерации с его помощью датасета, который в дальнейшем использовался при проведении сравнительного анализа библиотек.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Документация kotlin [Электронный ресурс] URL: <https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/> (Дата обращения: 20.09.2022)
2. Документация Material Design 3[Электронный ресурс] URL: <https://m3.material.io/> (Дата обращения: 15.09.2022)
3. Документация OData API в 1C[Электронный ресурс] URL: <https://1cfresh.com/articles/data_odata> (Дата обращения: 30.09.2022)
4. Документация Android[Электронный ресурс] URL:  
   <https://developer.android.com/> (Дата обращения: 16.10.2022)
5. Введение в паттер Backend-for-Frontend[Электронный ресурс] URL:   
   <https://medium.com/mobilepeople/backend-for-frontend-pattern-why-you-need-to-know-it-46f94ce420b0> (Дата обращения: 26.11.2022)
6. Установщик Treepoem [Электронный ресурс] URL: https://pypi.org/project/treepoem/ (Дата обращения: 26.10.2022)
7. Документация Aspose-barcode [Электронный ресурс] URL: <https://docs.aspose.com/barcode/python-net/product-features/#generate-barcodes> (Дата обращения: 16.10.2022)

## **ПРИЛОЖЕНИЕ 1. ТЗ**

1) Название проекта: Сравнение библиотек сканирования штрихкодов для Android и iOS.

2) Цель (назначение): Исследовать существующие библиотеки сканирования штрихкодов для мобильных устройств, провести сравнение и анализ используя сгенерированные синтетические данные.

3) Сроки выполнения: Начало - 2024-11-11, Конец - 2024-12-17

4) Исполнитель проекта (руководитель проекта): Комаров Георгий Юрьевич

5) Термины и сокращения:

* Штрихкод - графическая информация, наносимая на поверхность, маркировку или упаковку изделий, предоставляющая возможность считывания её техническими средствами — последовательность чёрных и белых полос, либо других геометрических фигур,
* 1D штрихкод - одномерный штрихкод, представляющий собой набор чередующихся полос и пробелов различной ширины,
* 2D штрихкод - двумерный штрихкод, представляющий собой графический символ в виде матрицы, где информация закодирована как в вертикальном, так и в горизонтальном направлениях,
* Дефект штрихкода - искажение или повреждение элементов штрихкода, которое затрудняет или делает невозможным его корректное считывание.

6) Технические требования.

* Инструмент для генерации штрих-кодов с наложением фильтра: язык разработки – Python,
* Инструмент для распознавания штрих-кодов на базе существующих библиотек – язык разработки Java/Android.

7) Содержание работы.

Этапы работы:

* Анализ предметной области,
* Определение методов исследования,
* Разработка скрипта для генерации штрихкодов,
* Разработка приложения для распознавания штрихкода на изображении используя библиотеки сканирования штрихкодов,
* Сравнительный анализ.

Задачи:

* Изучить теорию штрихкодирования, ответственный – Нгуен Хыу Жанг,
* Определить часто используемые форматы штрихкодов в различных отраслях, ответственный – Слиозберг Владимир Владимирович,
* Исследовать популярные библиотеки для сканирования штрихкодов для мобильных устройств, ответственный – Нгуен Хыу Жанг,
* Определить методы сравнения библиотек, ответственный – Слиозберг Владимир Владимирович,
* Разработать метрики для оценки производительности и точности библиотек, ответственный – Слиозберг Владимир Владимирович,
* Подготовить критерии для анализа библиотек по устойчивости к дефектам, ответственный – Нгуен Хыу Жанг,
* Изучить IDE Android Studio / IntelliJ IDEA, ответственный – Черепня Ярослав Игоревич,
* Изучить работу с изображениями в Java/Android, ответственный – Розметов Джалолиддин,
* Разработать инструмент для генерации штрихкодов без дефектов, ответственный – Маслов Иван Андреевич,
* Реализовать фильтры для наложения различных дефектов на изображение со штрихкодом, ответственный – Абакунов Кирилл Вячеславович,
* Разработать приложение для сканирования штрихкодов с использованием библиотеки ZXing, ответственный – Черепня Ярослав Игоревич,
* Разработать приложение для сканирования штрихкодов с использованием библиотеки Google ML Kit, ответственный – Розметов Джалолиддин,
* Разработать приложение для сканирования штрихкодов с использованием библиотеки Dynamsoft Barcode Reader, ответственный – Черепня Ярослав Игоревич,
* Разработать приложение для сканирования штрихкодов с использованием библиотеки Scandit, ответственный – Розметов Джалолиддин,
* Провести сравнительный анализ open-source библиотек, оформить отчёт с результатами, ответственные – Нгуен Хыу Жанг, Лиозберг Владимир Владимирович.

8) Основные результаты работы.

Результатом работы является таблица, где столбцы отображают сравниваемые библиотеки, а строки - критерии оценивания. В ячейках должны быть значения в виде чисел, которые отражают результат сравнения в рамках критерия. Также необходимо сделать выводы по полученным результатам и сформулировать рекомендации по внедрению оптимального решения.